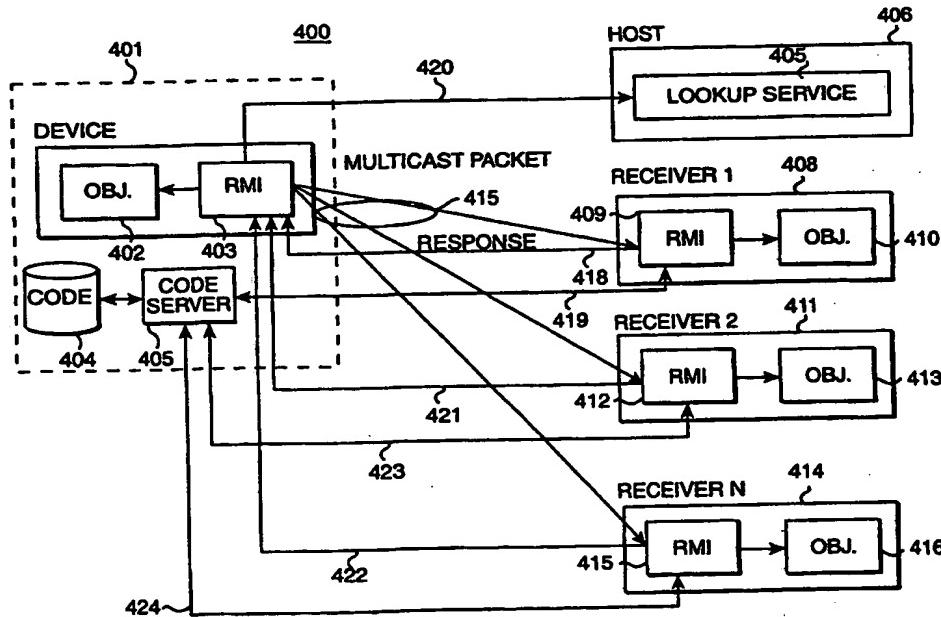




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>6</sup> :	A1	(11) International Publication Number: WO 99/44124
G06F 9/46		(43) International Publication Date: 2 September 1999 (02.09.99)
(21) International Application Number: PCT/US99/03223		
(22) International Filing Date: 16 February 1999 (16.02.99)		
(30) Priority Data:		
60/076,048 26 February 1998 (26.02.98) US		
09/044,939 20 March 1998 (20.03.98) US		
(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, MS UPAL01-521, Palo Alto, CA 94303 (US).		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
(72) Inventors: O'SULLIVAN, Bryan; 410 Broderick Street, San Francisco, CA 94117 (US). SCHEIFLER, Robert; 96 North Street, Somerville, MA 02144 (US). JONES, Peter, C.; 85 Bacon Street, Winchester, MA 01890 (US). WOLLRATH, Ann, M.; 9 Northwoods Road, Groton, MA 01450 (US). ARNOLD, Kenneth, C.R.C.; 7 Moon Hill Road, Lexington, MA 02173 (US). WALDO, James, H.; 155 Ruby Road, Dracut, MA 01826 (US).		
(74) Agents: GARRETT, Arthur S.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US) et al.		

(54) Title: APPARATUS AND METHOD FOR PROVIDING DOWNLOADABLE CODE FOR USE IN COMMUNICATING WITH A DEVICE IN A DISTRIBUTED SYSTEM



## (57) Abstract

Downloading code for communicating with a device that joins a network. When the device joins the network, it transmits a multicast packet including a reference to code for use in communicating with the device. Receivers in the network use the reference to download the code for constructing objects to communicate with the device, and the receivers respond with a reference to a lookup service in the network. Using the reference from the receivers, the device registers with the lookup service.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		

**APPARATUS AND METHOD FOR PROVIDING  
DOWNLOADABLE CODE FOR USE IN COMMUNICATING  
WITH A DEVICE IN A DISTRIBUTED SYSTEM**

**REFERENCE TO RELATED APPLICATIONS**

The following identified U.S. patent applications are relied upon and are incorporated by reference in this application as if fully set forth.

Provisional U.S. Patent Application No. 60/076,048, entitled "Distributed Computing System," filed on February 26, 1998.

U.S. Patent Application No. 09/044,923, entitled "Method and System for Leasing Storage," bearing attorney docket no. 06502.0011-01000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,838, entitled "Method, Apparatus, and Product for Leasing of Delegation Certificates in a Distributed System," bearing attorney docket no. 06502.0011-02000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,834, entitled "Method, Apparatus and Product for Leasing of Group Membership in a Distributed System," bearing attorney docket no. 06502.0011-03000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,916, entitled "Leasing for Failure Detection," bearing attorney docket no. 06502.0011-04000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,933, entitled "Method for Transporting Behavior in Event Based System," bearing attorney docket no. 06502.0054-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,919, entitled "Deferred Reconstruction of Objects and Remote Loading for Event Notification in a Distributed System," bearing attorney docket no. 06502.0062-01000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,938, entitled "Methods and Apparatus for Remote Method Invocation," bearing attorney docket no. 06502.0102-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/045,652, entitled "Method and System for Deterministic Hashes to Identify Remote Methods," bearing attorney docket no. 06502.0103-00000, and filed on the same date herewith.

**BEST AVAILABLE COPY**

U.S. Patent Application No. 09/044,790, entitled "Method and Apparatus for Determining Status of Remote Objects in a Distributed System," bearing attorney docket no. 06502.0104-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,930, entitled "Downloadable Smart Proxies for Performing Processing Associated with a Remote Procedure Call in a Distributed System," bearing attorney docket no. 06502.0105-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,917, entitled "Suspension and Continuation of Remote Methods," bearing attorney docket no. 06502.0106-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,835, entitled "Method and System for Multi-Entry and Multi-Template Matching in a Database," bearing attorney docket no. 06502.0107-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,839, entitled "Method and System for In-Place Modifications in a Database," bearing attorney docket no. 06502.0108, and filed on the same date herewith.

U.S. Patent Application No. 09/044,945, entitled "Method and System for Typesafe Attribute Matching in a Database," bearing attorney docket no. 06502.0109-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,931, entitled "Dynamic Lookup Service in a Distributed System," bearing attorney docket no. 06502.0110-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,826, entitled "Method and System for Facilitating Access to a Lookup Service," bearing attorney docket no. 06502.0113-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,932, entitled "Apparatus and Method for Dynamically Verifying Information in a Distributed System," bearing attorney docket no. 06502.0114-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/030,840, entitled "Method and Apparatus for Dynamic Distributed Computing Over a Network," bearing attorney docket no. 06502.0115-00000, and filed on February 26, 1998.

U.S. Patent Application No. 09/044,936, entitled "An Interactive Design Tool for Persistent Shared Memory Spaces," bearing attorney docket no. 06502.0116-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,934, entitled "Polymorphic Token-Based Control," bearing attorney docket no. 06502.0117-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,915, entitled "Stack-Based Access Control," bearing attorney docket no. 06502.0118-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,944, entitled "Stack-Based Security Requirements," bearing attorney docket no. 06502.0119-00000, and filed on the same date herewith.

U.S. Patent Application No. 09/044,837, entitled "Per-Method Designation of Security Requirements," bearing attorney docket no. 06502.0120-00000, and filed on the same date herewith.

#### FIELD OF THE INVENTION

The present invention relates to a system and method for transmitting objects between machines in a distributed system and more particularly to transmission of downloadable code for use in communicating with a device in a distributed system.

#### BACKGROUND OF THE INVENTION

Machines in a distributed system may communicate and access each other. When a device wants to join the distributed system, the machines already in the system typically must know how to communicate with the device. This requirement means that, for example, the machines must store code, drivers, and port information for any device that may join the system. In addition, they may be required to store stubs for communicating with any device that may join the system. A proxy stub is a reference to a remote object for use in reconstructing an object, and stubs facilitate communication between machines in a distributed system. These requirements may result in inefficiencies and increase memory or storage requirements due to the amount of prestored information required to communicate with devices that may join the system. Accordingly, a need exists for increased flexibility in transmitting code for communication between machines in a distributed system.

#### SUMMARY OF THE INVENTION

A first method consistent with the present invention transmits from a machine a multicast packet including a request to join a distributed system and including an indication of code for communicating with the machine. A response is received, the response including a reference to an entity for registration with the system, and a message is transmitted to the entity to register with the system.

A second method consistent with the present invention receives from a machine a multicast packet including a request to join a distributed system and including indication of code for communicating with the machine. The code is used to construct a response including a reference to an entity for registration with the system, and the response is transmitted to the machine.

A third method consistent with the present invention transmits from a first machine a first object including indication of code for communicating with the first machine, and a second machine receives the first object. The code is used for constructing a second object for communicating with the first machine, and the second object is transmitted from the second machine to the first machine.

A first apparatus consistent with the present invention transmits from a machine a multicast packet including a request to join a distributed system and including indication of code for communicating with the machine. The apparatus receives a response including a reference to an entity for registration with the system and transmits a message to the entity to register with the system.

A second apparatus consistent with the present invention receives from a machine a multicast packet including a request to join a distributed system and including indication of code for communicating with the machine. The apparatus uses the code to construct a response including a reference to an entity for registration with the system and transmits the response to the machine.

A third apparatus consistent with the present invention transmits from a first machine a first object including indication of code for communicating with the first machine. The apparatus receives at a second machine the first object and uses the code for constructing a second object for communicating with the first machine. The second object is transmitted from the second machine to the first machine.

#### BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings are incorporated in and constitute a part of this specification and, together with the description, explain the advantages and principles of the invention. In the drawings,

FIG. 1 is a diagram of an exemplary distributed processing system that can be used in an implementation consistent with the present invention;

FIG. 2 is a diagram of an exemplary distributed system infrastructure;

FIG. 3 is a diagram of a computer in a distributed system infrastructure shown in FIG. 2;

FIG. 4 is a diagram of an exemplary distributed network illustrating a device joining the network; and

FIG. 5 is a flow chart of a process for a device to join a distributed network.

## DETAILED DESCRIPTION

### Overview

Methods and systems consistent with the present invention provide a technique for joining a device to a distributed network. When the device joins the network, it transmits a multicast packet that includes code or a reference to code for use in communicating with the device. Receivers in the network use the reference to download the code to communicate with the device. The receivers thus need not have prestored code or other information required to communicate with the device, which may enhance network efficiency.

The receivers respond to the multicast packet by transmitting to the device a reference to a lookup service in the network. Using the reference from at least one of the receivers, the device registers itself with the lookup service. A lookup service is described in U.S. patent application filed on even date herewith, assigned to a common assignee, and entitled "Method and System for Facilitating Access to a Lookup Service," which is incorporated herein by reference.

### Distributed Processing System

FIG. 1 illustrates an exemplary distributed processing system 100 which can be used in an implementation consistent with the present invention. In FIG. 1, distributed processing system 100 contains three independent and heterogeneous platforms 101, 102, and 103 connected in a network configuration represented by network cloud 119. The composition

and protocol of the network configuration represented by cloud 119 is not important as long as it allows for communication of the information between platforms 101, 102 and 103. In addition, the use of just three platforms is merely for illustration and does not limit an implementation consistent with the present invention to the use of a particular number of platforms. Further, the specific network architecture is not crucial to embodiments consistent with the present invention. For example, another network architecture that could be used in an implementation consistent with this invention would employ one platform as a network controller to which all the other platforms would be connected.

In the implementation of distributed processing system 100, platforms 101, 102 and 103 each include a processor 116, 117, and 118 respectively, and a memory, 104, 105, and 106, respectively. Included within each memory 104, 105, and 106, are applications 107, 108, and 109, respectively, operating systems 110, 111, and 112, respectively, and Java™ remote method invocation components (RMI) components 113, 114, and 115, respectively. RMI is explained in, for example, the following document, which is incorporated herein by reference: Remote Method Invocation Specification, Sun Microsystems, Inc. (1997), which is available via universal resource locator (URL)

<http://www.javasoft.com/products/jdk/1.1/docs/guide/rmi/spec/rmiTOC.doc.html>.

Applications 107, 108, and 109 can be programs that are either previously written and modified to work with, or that are specially written to take advantage of, the services offered by an implementation consistent with the present invention. Applications 107, 108, and 109 invoke operations to be performed in accordance with an implementation consistent with this invention.

Operating systems 110, 111, and 112 are typically standard operating systems tied to the corresponding processors 116, 117, and 118, respectively. The platforms 101, 102, and 103 can be heterogenous. For example, platform 101 has an UltraSparc® microprocessor manufactured by Sun Microsystems, Inc. as processor 116 and uses a Solaris® operating system 110. Platform 102 has a MIPS microprocessor manufactured by Silicon Graphics Corp. as processor 117 and uses a Unix operating system 111. Finally, platform 103 has a Pentium microprocessor manufactured by Intel Corp. as processor 118 and uses a Microsoft Windows 95 operating system 112. An implementation consistent with the present invention is not so limited and could accommodate homogenous platforms as well.

Sun, Sun Microsystems, Solaris, Java, and the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UltraSparc and all other SPARC trademarks are used under license and are trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

Memories 103, 104, and 105 serve several functions, such as general storage for the associated platform. Another function is to store applications 107, 108, and 109, RMI components 113, 114, and 115, and operating systems 110, 111, and 112 during execution by the respective processor 116, 117, and 118. In addition, portions of memories 103, 104, and 105 may constitute shared memory available to all of the platforms 101, 102, and 103 in network 119. Note that RMI components 113, 114, and 115 operate in conjunction with a JVM, which is not shown for the purpose of simplifying the figure.

#### Distributed System Infrastructure

Systems and methods consistent with the present invention may also operate within a particular distributed system 200, which will be described with reference to FIGS. 2 and 3. This distributed system 200 is comprised of various components, including hardware and software, to (1) allow users of the system to share services and resources over a network of many devices; (2) provide programmers with tools and programming patterns that allow development of robust, secured distributed systems; and (3) simplify the task of administering the distributed system. To accomplish these goals, distributed system 400 utilizes the Java programming environment to allow both code and data to be moved from device to device in a seamless manner. Accordingly, distributed system 200 is layered on top of the Java programming environment and exploits the characteristics of this environment, including the security offered by it and the strong typing provided by it.

In distributed system 200 of FIGS. 2 and 3, different computers and devices are federated into what appears to the user to be a single system. By appearing as a single system, distributed system 200 provides the simplicity of access and the power of sharing that can be provided by a single system without giving up the flexibility and personalized response of a personal computer or workstation. Distributed system 200 may contain thousands of devices operated by users who are geographically disperse, but who agree on basic notions of trust, administration, and policy.

Within an exemplary distributed system are various logical groupings of services provided by one or more devices, and each such logical grouping is known as a Djinn. A "service" refers to a resource, data, or functionality that can be accessed by a user, program, device, or another service and that can be computational, storage related, communication related, or related to providing access to another user. Examples of services provided as part of a Djinn include devices, such as printers, displays, and disks; software, such as programs or utilities; information, such as databases and files; and users of the system.

Both users and devices may join a Djinn. When joining a Djinn, the user or device adds zero or more services to the Djinn and may access, subject to security constraints, any one of the services it contains. Thus, devices and users federate into a Djinn to share access to its services. The services of the Djinn appear programmatically as objects of the Java programming environment, which may include other objects, software components written in different programming languages, or hardware devices. A service has an interface defining the operations that can be requested of that service, and the type of the service determines the interfaces that make up that service.

Distributed system 200 is comprised of computer 202, a computer 204, and a device 206 interconnected by a network 208. Device 206 may be any of a number of devices, such as a printer, fax machine, storage device, computer, or other devices. Network 208 may be a local area network, wide area network, or the Internet. Although only two computers and one device are depicted as comprising distributed system 200, one skilled in the art will appreciate that distributed system 200 may include additional computers or devices.

FIG. 3 depicts computer 202 in greater detail to show a number of the software components of distributed system 200. One skilled in the art will appreciate that computer 204 or device 206 may be similarly configured. Computer 202 includes a memory 302, a secondary storage device 304, a central processing unit (CPU) 306, an input device 308, and a video display 310. Memory 302 includes a lookup service 312, a discovery server 314, and a Java runtime system 316. The Java runtime system 316 includes the Java RMI system 318 and a JVM 320. Secondary storage device 304 includes a Java space 322.

As mentioned above, distributed system 200 is based on the Java programming environment and thus makes use of the Java runtime system 316. The Java runtime system 316 includes the Java API libraries, allowing programs running on top of the Java runtime

system to access, in a platform-independent manner, various system functions, including windowing capabilities and networking capabilities of the host operating system. Since the Java API libraries provides a single common API across all operating systems to which the Java runtime system is ported, the programs running on top of a Java runtime system run in a platform-independent manner, regardless of the operating system or hardware configuration of the host platform. The Java runtime system 316 is provided as part of the Java software development kit available from Sun Microsystems, Inc. of Mountain View, CA.

JVM 320 also facilitates platform independence. JVM 320 acts like an abstract computing machine, receiving instructions from programs in the form of bytecodes and interpreting these bytecodes by dynamically converting them into a form for execution, such as object code, and executing them. RMI 318 facilitates remote method invocation by allowing objects executing on one computer or device to invoke methods of an object on another computer or device. Both RMI and the JVM are also provided as part of the Java software development kit.

Lookup service 312 defines the services that are available for a particular Djinn. That is, there may be more than one Djinn and, consequently, more than one lookup service within distributed system 200. Lookup service 312 contains one object for each service within the Djinn, and each object contains various methods that facilitate access to the corresponding service. Lookup service 312 is described in U.S. patent application entitled "Method and System for Facilitating Access to a Lookup Service," which was previously incorporated herein by reference.

Discovery server 314 detects when a new device is added to distributed system 200, during a process known as boot and join (or discovery), and when such a new device is detected, the discovery server passes a reference to lookup service 312 to the new device so that the new device may register its services with the lookup service and become a member of the Djinn. After registration, the new device becomes a member of the Djinn, and as a result, it may access all the services contained in lookup service 312. The process of boot and join is described in U.S. patent application entitled "Apparatus and Method for providing Downloadable Code for Use in Communicating with a Device in a Distributed System," which was previously incorporated herein by reference.

A Java space 322 is an object repository used by programs within distributed system 200 to store objects. Programs use a Java space 322 to store objects persistently as well as to make them accessible to other devices within distributed system 200. Java spaces are described in U.S. patent application serial no. 08/971,529, entitled "Database System Employing Polymorphic Entry and Entry Matching," assigned to a common assignee, and filed on November 17, 1997, which is incorporated herein by reference. One skilled in the art will appreciate that an exemplary distributed system 200 may contain many lookup services, discovery servers, and Java spaces.

#### Data Flow in a Distributed Processing System

FIG. 4 is a diagram of an exemplary object-oriented distributed network or system 400 connecting machines 401, 408, 411, and 414, such as computers or virtual machines executing on one or more computers, or the machines described with reference to FIGS. 1, 2, and 3. As shown in FIG. 4, transmitting machine or device 401 includes an object 402 and RMI 403 for performing processing on objects. Device 401 intends to join distributed network 400 including other devices or machines, referred to as receivers. Network 400 includes a host 406 containing a lookup service 407 and also includes receivers 408, 411, and 414. Only three receivers are shown for illustrative purposes only; network 400 may include fewer or more receivers and would usually include many receivers. Each receiver shown contains RMI, shown by RMIs 409, 412, and 415. Each of the RMIs for the receivers constructs a remote reference to object 402. These remote references include objects 410, 413, and 416 for, respectively, receivers 408, 411, and 414.

In order to join network 400, device 401 transmits a multicast packet 417 including a an indication of code 404, such as a reference to the code or the code itself, required to communicate with device 401. A multicast packet is a message sent to a plurality of devices in a network. Device 401 uses RMI 403 to provide flexibility, allowing code to be moved when necessary along with information or the object's state and type. Additionally, device 401 may include in multicast packet 417 an identification of the type of object transmitted, the data constituting the state of the object, and a network-accessible location in the form of a URL for code that is associated with the object. Multicast packet 417 may also include a stub for communicating with the booting device. URLs are known in the art and an explanation, which is incorporated herein by reference, appears in, for example, a text entitled "The Java

Tutorial: Object-Oriented Programming for the Internet," pp. 494-507, by Mary Campione and Kathy Walrath, Addison-Wesley, 1996.

In response to multicast packet 417, receivers 408, 411, and 414 transmit responses 418, 421, and 422, each response including a reference for use in communicating with lookup service 407. When receiving machine 408 receives multicast packet 417, it identifies the type of transmitted object. In order to send the response and communicate with device 401, receivers 408, 411, and 414 use the reference from device 401 to transmit calls 419, 423, and 424 to code server 405 in order to obtain code 404 for communicating with device 401. A code server is an entity and process that has access to code and responds to requests for a particular type or class of object and returns code for that object. A code server may be located within device 401 or on another machine. Also, the code may be resident on the same platform as the code server or on a separate platform.

RMI 409, for example, may use a URL from multicast packet 417 to locate the code and transfer a copy of the code to receiver 408. Because the code is bytecodes and is therefore portable, a receiving machine can load the code into its RMI to reconstruct the object. Thus, receivers 408, 411, and 414 can reconstruct an object of the appropriate type even if that kind of object has not been present on the machines before.

Receivers 408, 411, and 414 use code 404 to construct an object for transmitting responses 418, 421, and 422. Alternatively, device 401 may transmit the code instead of a reference to the code. Device 401 receives the responses and, using at least one of the transmitted references, registers itself with lookup service 407 via call 420. These calls may be remote procedure calls, and the objects may be transmitted as byte streams. Device 401 may transmit a message to register with a lookup service, the message containing information identifying the device so that machines using the lookup service may access or communicate with the device. Remote procedure calls, and methods for downloading code, are described in U.S. patent application serial no. 08/950,756, filed on October 15, 1997, and entitled "Deferred Reconstruction of Objects and Remote Loading in a Distributed System," which is incorporated herein by reference. Streams used in the Java programming language, including input and output streams, are known in the art and an explanation, which is incorporated herein by reference, appears in, for example, a text entitled "The Java<sup>TM</sup> Tutorial: Object-

Oriented Programming for the Internet," pp. 325-53, by Mary Campione and Kathy Walrath, Addison-Wesley, 1996.

#### Process for a Device to Join a Network

FIG. 5 is a flow chart of a process 500 for device to join a network, such as distributed network 400. A device wishing to join a network transmits a multicast packet to the network, the packet including an indication of code, such as a reference to code or the code itself, for use in communicating with the device (step 501). Table 1 shows in pseudocode an example of information in a multicast packet. This packet may be transmitted from what is referred to as a bootstrap or discovery request service, which runs on the discovery server identified above with reference to FIG. 3 and accepts requests for references to a lookup service. Therefore, when a device joins a network in this example, referred to as "boot" into a Djinn, it may use the bootstrap request service to boot into the network.

Table 1

bootstrap protocol version transport-specific endpoint info device ID URL from which to load stub name of the stub class	<pre> int protoVersion; Object endpoint; String id = DeviceID.getDevice ID (); java.net.URL codebase; String stubType;  ByteArrayOutputStream byteStr = new ByteArrayOutputStream ( ); ObjectOutputStream objStr = new ObjectOutputStream (byteStr);  objStr.writeInt (protoVersion); objStr.writeObject (endpoint); objStr.writeUTF (id); objStr.writeUTF (codebase.toExternalForm ( ) ); objStr.writeUTF (stubType); </pre> <p>the final product</p> <pre>byte [ ] packetBody = byteStr.toByteArray ( );</pre>
--	--

The following is an explanation of the fields of the exemplary packet shown in Table

1. The "protoVersion" field contains an integer which indicates the version of the bootstrap protocol in operation for a device to join a network. The "endpoint" field encapsulates transport-dependent endpoint information which allows instances of the bootstrap request service to construct valid references to the bootstrap response service on the booting entity.

The "id" field contains a "Device ID" string, which identifies the booting device. The "codebase" field provides a URL from which instances of the bootstrap request service can download stub code for the bootstrap response service. The "stubType" field provides a fully-qualified name of the stub class for the bootstrap response service.

Upon receiving the multicast packet receivers in the network download the code to construct an object for communicating with the device. The receivers each transmit the object to the device, the object including a reference to a lookup service in the network (step 502). The receivers may use what is referred to as a bootstrap or discovery response service, which runs on a discovering (booting) entity and accepts references to the lookup service.

Table 2 illustrates an exemplary Java interface for a bootstrap response service.

Table 2

```
public interface ResponseService extends java.rmi.Remote
{
    public void advertiseLookup (ServiceRegistrar lookup)
        throws java.rmi.RemoteException;
}
```

The receivers, through the bootstrap response service, provide a reference to an object that implements a service registrar, which may correspond to the lookup service explained in U.S. patent application filed on even date herewith, assigned to a common assignee, and entitled "Dynamic Lookup Service for a Distributed System," which is incorporated herein by reference. If the device received no response to the multicast packet, it may continue sending multicast packets at regular intervals.

Upon receiving the object, the device uses at least one of the reference transmitted by the receivers in order to register with the lookup service (step 503) through the service registrar. After registration, the lookup service subsequently contains an identification of the device, and it is thus accessible to other entities in the network. In addition to registering with the lookup service, the device may optionally respond, providing notification of its registration or other information, to each of the receivers from which it received a response to the multicast packet (step 504).

The protocol and method described above may also be used for providing an object containing all behavior necessary to communicate with a particular device. Thus, devices do not necessarily require that they maintain an indication of code for communicating with the

particular device, as they may obtain that information from the particular device using the method described above.

Machines implementing the steps shown in FIG. 5 may include computer processors for performing the functions, as shown in FIGS. 1, 2, 3, and 4. They may include modules or programs configured to cause the processors to perform the above functions. They may also include computer program products stored in a memory. The computer program products may include a computer-readable medium or media having computer-readable code embodied therein for causing the machines to perform functions described above. The media may include a computer data signal embodied in a carrier wave and representing sequences of instructions which, when executed by a processor, cause the processor to securely address a peripheral device at an absolute address by performing the method described in this specification. The media may also include data structures for use in performing the method described in this specification.

Although the illustrative embodiments of the systems consistent with the present invention are described with reference to a computer system implementing the Java programming language on the JVM specification, the invention is equally applicable to other computer systems processing code from different programming languages. Specifically, the invention may be implemented with both object-oriented and nonobject-oriented programming systems. In addition, although an embodiment consistent with the present invention has been described as operating in the Java programming environment, one skilled in the art will appreciate that the present invention can be used in other programming environments as well.

While the present invention has been described in connection with an exemplary embodiment, it will be understood that many modifications will be readily apparent to those skilled in the art, and this application is intended to cover any adaptations or variations thereof. For example, different labels or definitions for the multicast packet may be used without departing from the scope of the invention. This invention should be limited only by the claims and equivalents thereof.

**WHAT IS CLAIMED IS:**

1. A method for transmitting objects in a distributed system comprised of multiple machines, comprising:

transmitting from a machine a multicast packet including a request to join the distributed system and including an indication of code for communicating with the machine;  
receiving a response including a reference to an entity for registration with the system;  
and  
transmitting a message to the entity to register with the system.

2. The method of claim 1 wherein the transmitting includes  
transmitting a reply to the response.

3. A method for processing objects in a distributed system comprised of multiple machines, comprising:

receiving from a machine a multicast packet including a request to join the distributed system and including an indication of code for communicating with the machine;  
using the code to construct a response including a reference to an entity for registration with the system; and  
transmitting the response to the machine.

4. The method of claim 3 wherein the receiving includes  
receiving a reply to the response.

5. A method for transmitting objects in a distributed system comprised of multiple machines, comprising:

transmitting from a first machine a first object including an indication of code for communicating with the first machine;  
receiving at a second machine the first object;  
using the code for constructing a second object for communicating with the first machine; and  
transmitting the second object from the second machine to the first machine.

6. An apparatus for transmitting objects in a distributed system comprised of multiple machines, comprising:

a module configured to transmit from a machine a multicast packet including a request to join the distributed system and including an indication of code for communicating with the machine;

a module configured to receive a response including a reference to an entity for registration with the system; and

a module configured to transmit a message to the entity to register with the system.

7. The apparatus of claim 6 wherein the transmit module includes

a module configured to transmit a reply to the response.

8. An apparatus for processing objects in a distributed system comprised of multiple machines, comprising:

a module configured to receive from a machine a multicast packet including a request to join the distributed system and including an indication of code for communicating with the machine;

a module configured to use code to construct a response including a reference to an entity for registration with the system; and

a module configured to transmit the response to the machine.

9. The apparatus of claim 8 wherein the receive module includes

a module configured to receive a reply to the response.

10. An apparatus for transmitting objects in a distributed system comprised of multiple machines, comprising:

a module configured to transmit from a first machine a first object including an indication of code for communicating with the first machine;

a module configured to receive at a second machine the first object;

a module configured to use the code for constructing a second object for communicating with the first machine; and

a module configured to transmit the second object from the second machine to the first machine.

11. A system for transmitting objects in a distributed system comprised of multiple machines, comprising:

a first machine;

a plurality of second machines;

a network connecting the first machine with the plurality of second machines; and an apparatus for joining the first machine with the network, the apparatus including:

a module configured to transmit from the first machine a multicast packet including a request to join the distributed system and including an indication of code for communicating with the first machine;

a module configured to receive a response including a reference to an entity for registration with the system; and

a module configured to transmit a message to the entity to register with the system.

12. The system of claim 11 wherein the transmit module includes a module configured to transmit a reply to the response.

13. A system for transmitting objects in a distributed system comprised of multiple machines, comprising:

a first machine;

a plurality of second machines;

a network connecting the first machine with the plurality of second machines; and an apparatus for joining the first machine with the network, the apparatus including:

a module configured to receive from the first machine a multicast packet including a request to join the distributed system and including an indication of code for communicating with the first machine;

a module configured to use the code to construct a response including a reference to an entity for registration with the system; and

a module configured to transmit the response to the first machine.

14. The system of claim 13 wherein the receive module includes a module configured to receive a reply to the response.
15. A system for transmitting objects in a distributed system comprised of multiple machines, comprising:
  - a first machine;
  - a second machine;
  - a network connecting the first machine with the plurality of second machines; and
  - an apparatus for communicating between the first machine and the second machine, the apparatus including:
    - a module configured to transmit from the first machine a first object including an indication of code for communicating with the first machine;
    - a module configured to receive at the second machine the first object;
    - a module configured to use the code for constructing a second object for communicating with the first machine; and
    - a module configured to transmit the second object from the second machine to the first machine.
16. A computer program product, comprising:
  - a computer-readable medium containing instructions for controlling a computer system to perform a method, the method including:
    - transmitting from a machine a multicast packet including a request to join the distributed system and including an indication of code for communicating with the machine;
    - receiving a response including a reference to an entity for registration with the system;
  - and
  - transmitting a message to the entity to register with the system.
17. The computer program product of claim 16 wherein the transmitting includes transmitting a reply to the response.

18. A computer program product, comprising:  
a computer-readable medium containing instructions for controlling a computer system to perform a method, the method including:  
receiving from a machine a multicast packet including a request to join the distributed system and including an indication of code for communicating with the machine;  
using the code to construct a response including a reference to an entity for registration with the system; and  
transmitting the response to the machine.
19. The computer program product of claim 18 wherein the receiving includes receiving a reply to the response.
20. A computer program product, comprising:  
a computer-readable medium containing instructions for controlling a computer system to perform a method, the method including:  
transmitting from a first machine a first object including an indication of code for communicating with the first machine;  
receiving at a second machine the first object;  
using the code for constructing a second object for communicating with the first machine; and  
transmitting the second object from the second machine to the first machine.
21. An article of manufacture specifying a representation of an object stored in a computer-readable storage medium and capable of electronic transmission between machines in a distributed system, the article of manufacture comprising:  
an object comprising a multicast packet to be transmitted to a plurality of machines in the system, the packet including a request by a machine to join the system and including an indication of code for use in communicating with the machine.

22. An article of manufacture specifying a representation of an object stored in a computer-readable storage medium and capable of electronic transmission between machines in a distributed system, the article of manufacture comprising:

an object to be transmitted from a first machine to a second machine in the system, the object including an indication of code for use in communicating with the first machine.

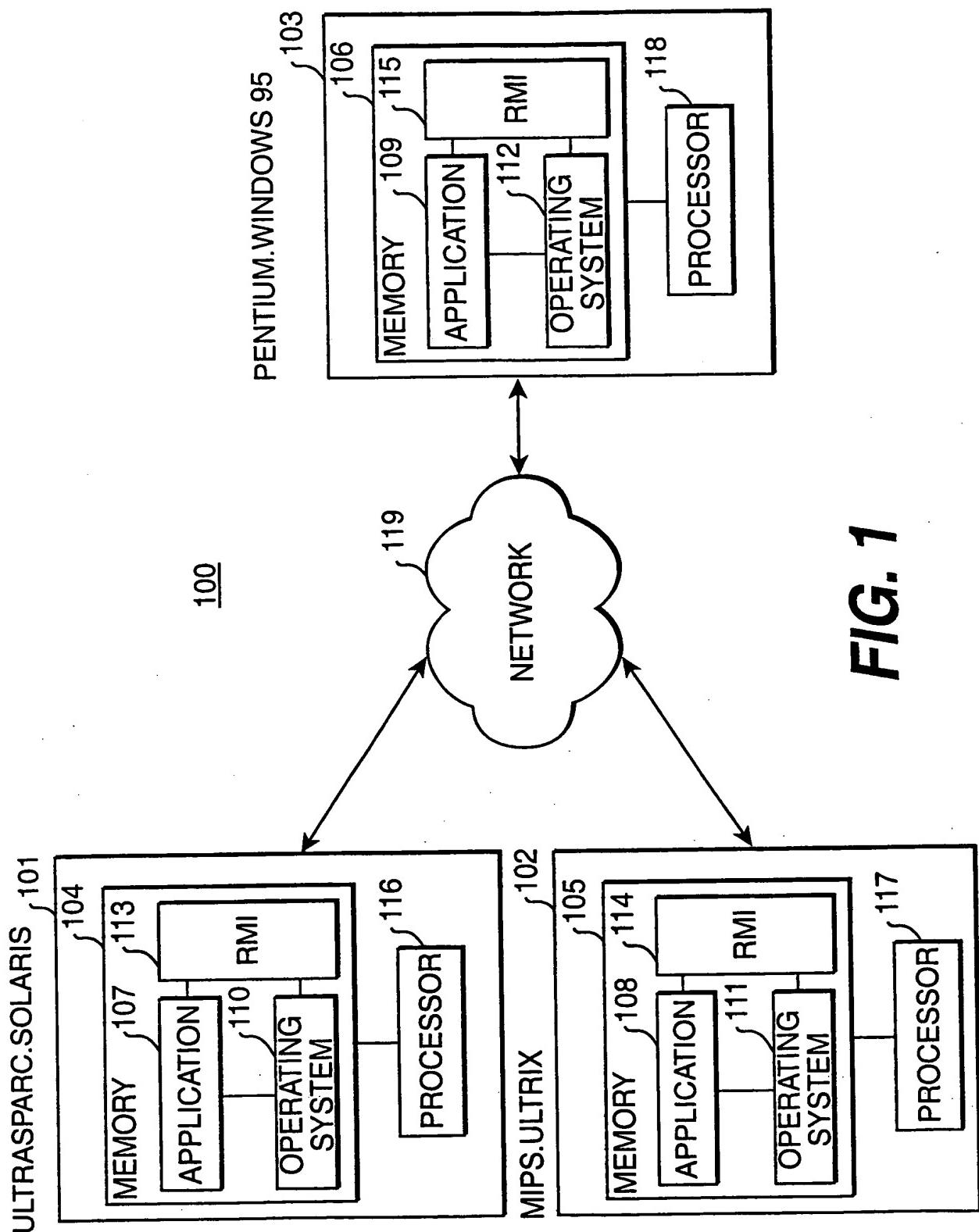
23. An apparatus for transmitting objects in a distributed system comprised of multiple machines, comprising:

means for transmitting from a machine a multicast packet including a request to join the system and including an indication of code for communicating with the machine;

means for receiving a response including a reference to an entity for registration with the system; and

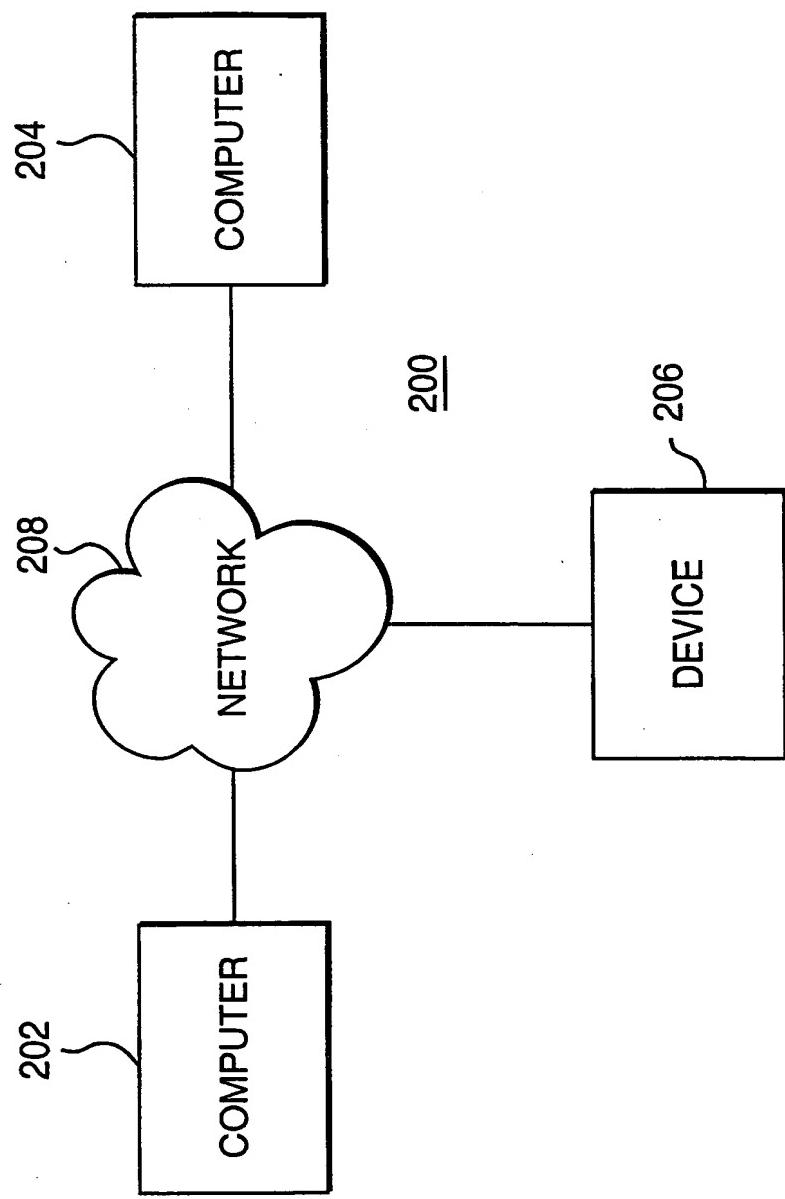
means for transmitting a message to the entity to register with the system.

1 / 5



SUBSTITUTE SHEET (RULE 26)

2 / 5

**FIG. 2**

SUBSTITUTE SHEET (RULE 26)

3 / 5

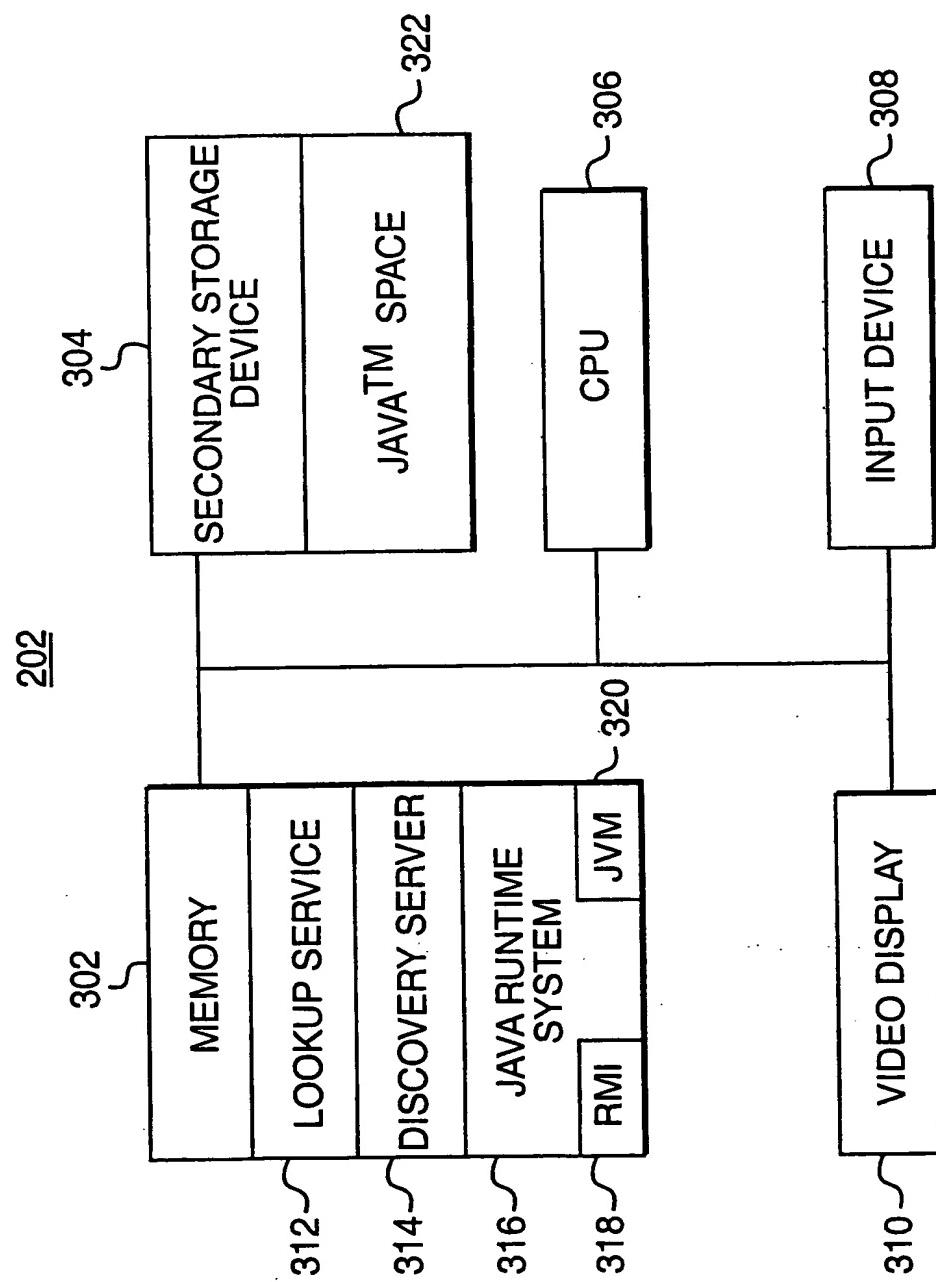
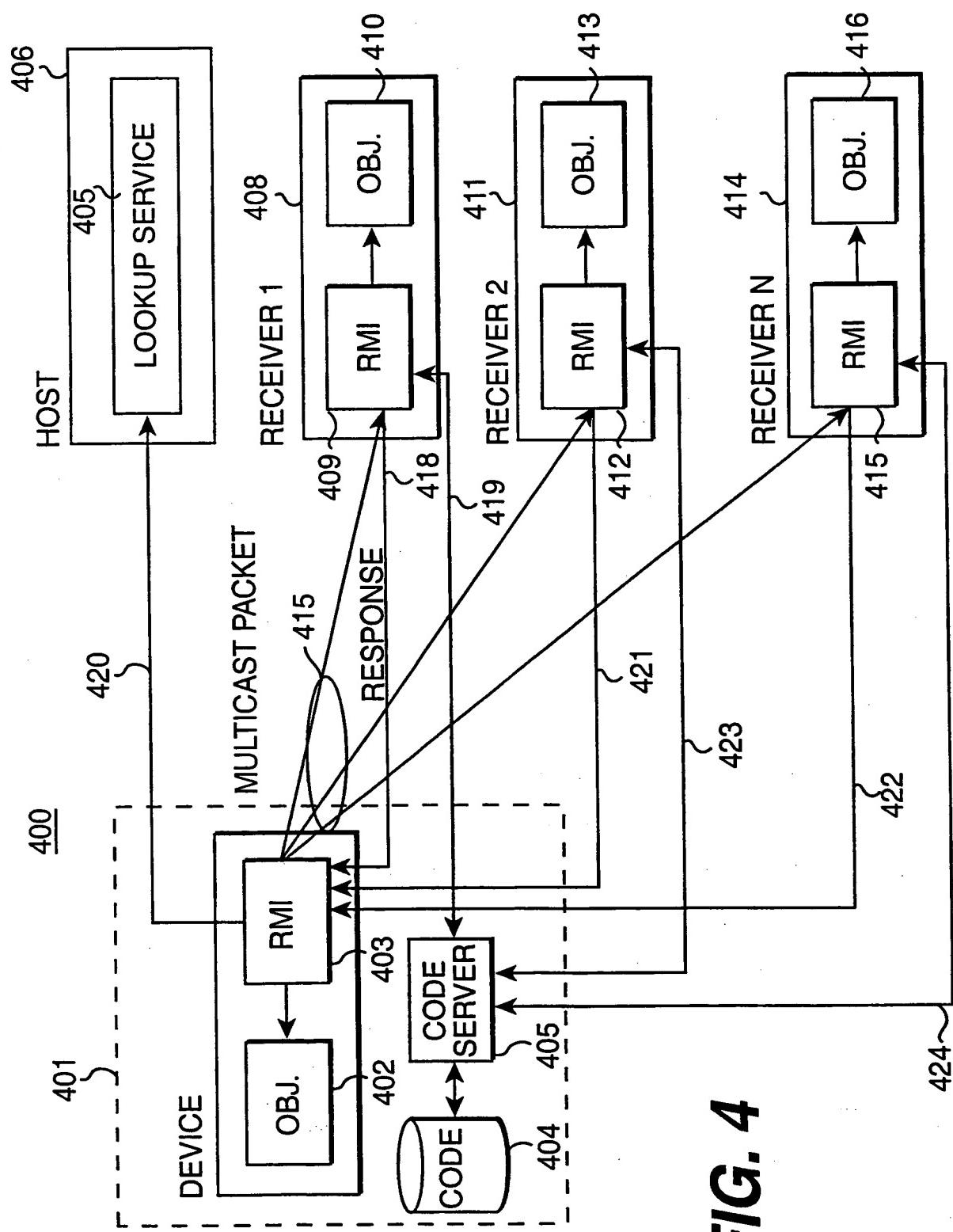


FIG. 3

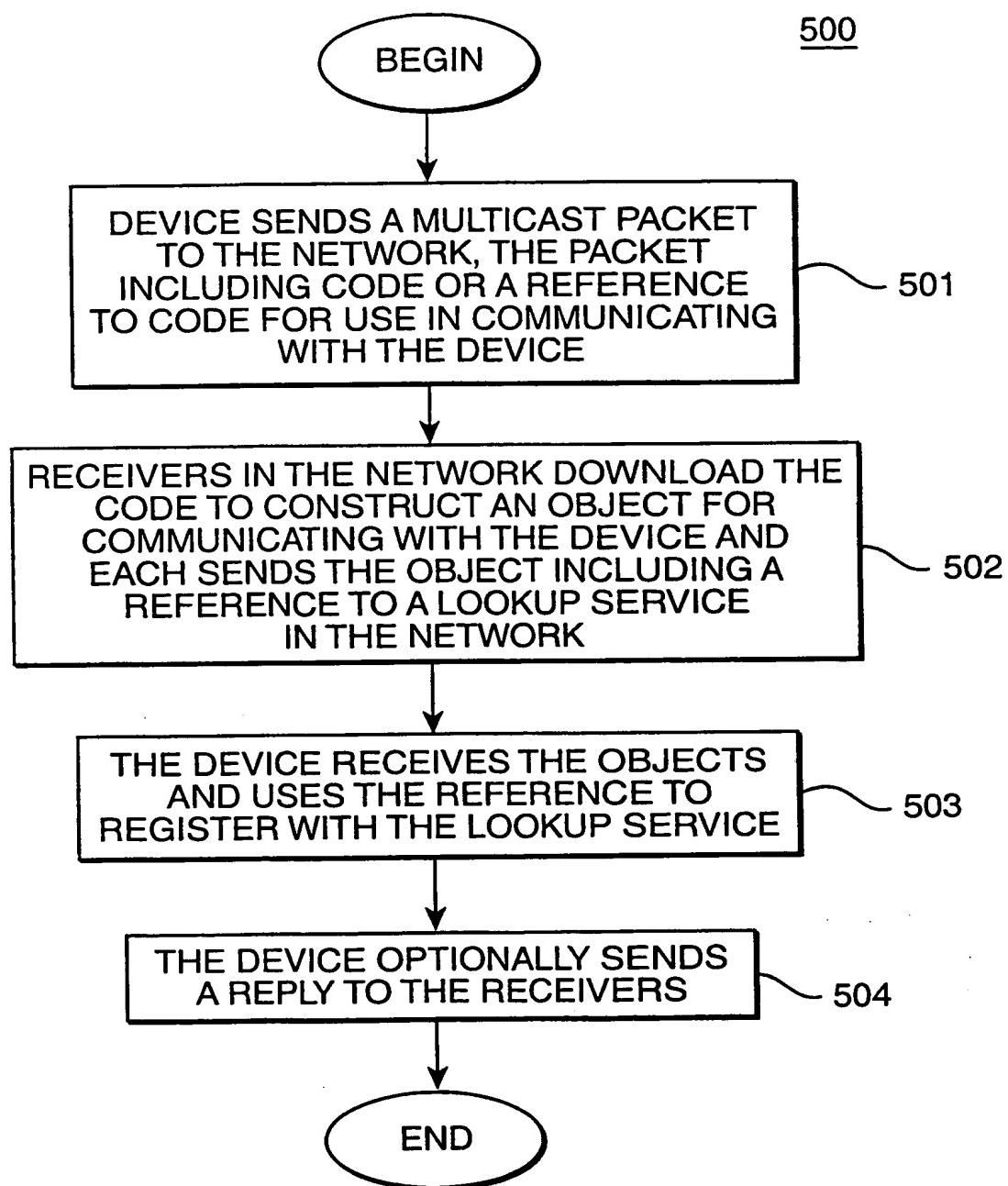
SUBSTITUTE SHEET (RULE 26)

4 / 5

**FIG. 4**

SUBSTITUTE SHEET (RULE 26)

5 / 5

**FIG. 5**

SUBSTITUTE SHEET (RULE 26)

# INTERNATIONAL SEARCH REPORT

International Application No  
PCT/US 99/03223

**A. CLASSIFICATION OF SUBJECT MATTER**  
**IPC 6 G06F9/46**

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)  
**IPC 6 H04L G06F**

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	GUTH, ROB: "JavaOne: Sun to expand Java distributed computing effort" HTTP://WWW.SUNWORLD.COM/SWOL-02-1998/SWOL-02-SUNSPOTS.HTML, 20 February 1998 (1998-02-20), page 1 XP002109935 USA the whole document	5,10,15, 20,22
A	---	1-4,6-9, 11-14, 16-19, 21,23
X	EP 0 472 874 A (IBM) 4 March 1992 (1992-03-04) column 5, line 7 - line 56	5,10,15, 20,22
A	---	1-4,6-9, 11-14, 16-19, 21,23
-/-		

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

\* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority, claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

26 July 1999

Date of mailing of the international search report

09/08/1999

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl.  
Fax: (+31-70) 340-3016

Authorized officer

Bijn, K

## INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/03223

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 4 823 122 A (MANN BRUCE ET AL) 18 April 1989 (1989-04-18) column 5, line 28 - column 7, line 27 -----	1-23
X	GB 2 262 825 A (INTEL CORP)	22
Y	30 June 1993 (1993-06-30) page 5, line 1 - page 6, line 8	1-4, 6-9, 11-14, 16-19, 21, 23
A	-----	5, 10, 15, 20
Y	EP 0 805 393 A (IBM) 5 November 1997 (1997-11-05)  column 1, line 10 - column 2, line 5 column 10, line 28 - column 11, line 15 -----	1-4, 6-9, 11-14, 16-19, 21, 23

# INTERNATIONAL SEARCH REPORT

Information on patent family members

national Application No

PCT/US 99/03223

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 0472874	A	04-03-1992	US 5446897 A DE 69118053 D DE 69118053 T JP 2070484 C JP 4227151 A JP 7110008 B	29-08-1995 25-04-1996 02-10-1996 10-07-1996 17-08-1992 22-11-1995
US 4823122	A	18-04-1989	US 5734659 A US 5621734 A US 4975905 A US 4975904 A AU 633510 B AU 4141689 A AU 633511 B AU 4141789 A AU 591057 B AU 4266185 A BR 8502706 A CA 1257399 A CA 1279933 A CA 1301941 A DE 3584853 A DE 3586430 A DE 3586431 A DE 3586433 A DE 3586434 A DE 3586633 A DE 3586634 A EP 0163577 A EP 0380141 A EP 0374131 A EP 0374132 A EP 0375664 A EP 0374133 A EP 0374134 A FI 852198 A,B, IE 57544 B JP 2515075 B JP 5063706 A JP 1922242 C JP 6048812 B JP 61056538 A JP 2698336 B JP 8214003 A US 5058108 A	31-03-1998 15-04-1997 04-12-1990 04-12-1990 04-02-1993 21-12-1989 04-02-1993 21-12-1989 30-11-1989 05-12-1985 12-02-1986 11-07-1989 05-02-1991 26-05-1992 23-01-1992 03-09-1992 03-09-1992 03-09-1992 03-09-1992 15-10-1992 15-10-1992 04-12-1985 01-08-1990 20-06-1990 20-06-1990 27-06-1990 20-06-1990 20-06-1990 02-12-1985 07-10-1992 10-07-1996 12-03-1993 07-04-1995 22-06-1994 22-03-1986 19-01-1998 20-08-1996 15-10-1991
GB 2262825	A	30-06-1993	US 5319751 A DE 4244266 A JP 5265919 A US 5404494 A US 5412798 A	07-06-1994 01-07-1993 15-10-1993 04-04-1995 02-05-1995
EP 0805393	A	05-11-1997	US 5787249 A US 5704032 A US 5699501 A US 5793962 A US 5696896 A JP 10040222 A	28-07-1998 30-12-1997 16-12-1997 11-08-1998 09-12-1997 13-02-1998

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

national Application No

PCT/US 99/03223

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0805393 A		JP 10040226 A	13-02-1998
		JP 10040227 A	13-02-1998
		JP 10040229 A	13-02-1998

---

**THIS PAGE BLANK (USPTO)**